Carnegie Mellon

Towards Rapid Re-Certification Using Formal Analysis

Daniel Smullen

Travis Breaux

Carnegie Mellon University

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
MAY 2015		**00-00-2015 to 00-00-2015**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Towards Rapid Re-Certification Using Formal Analysis	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Carnegie Mellon University,Institute of Software Research,Pittsburgh,PA,15213**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
Presented at the 12th Annual Acquisition Research Symposium held May 13-14, 2015 in Monterey, CA.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**35**	

Outline

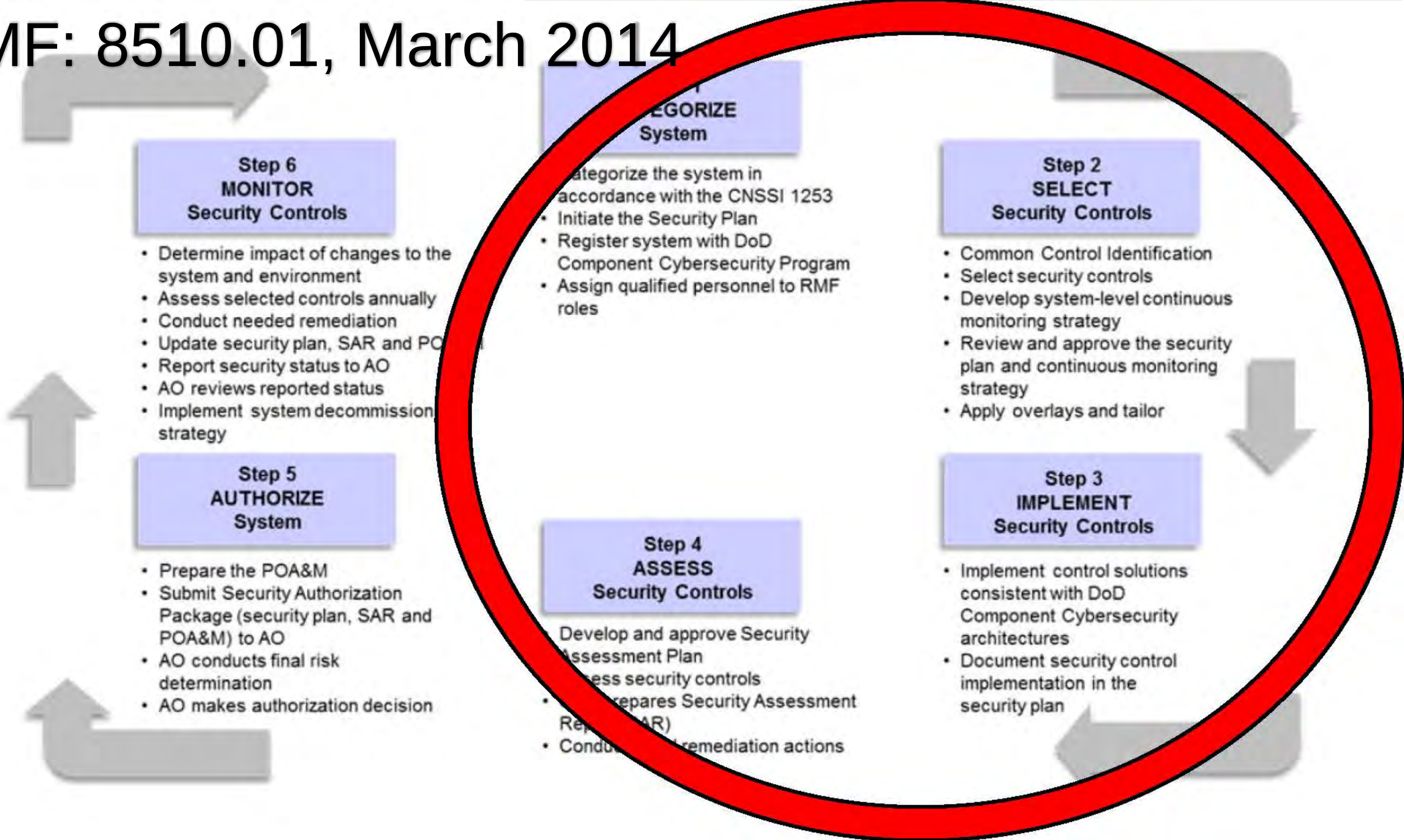
1. Problem Overview
   - Why is software (re)certification hard?
   - What's the risk?
2. What kind of solution is needed?
3. Technical Background
4. Approach, Running Example
   - Conflict Detection, Reconciliation
5. Recertification Triggers
6. Does it scale?
7. Future Work

Why is software (re)certification hard?

- Systems change, requirements evolve.
- As changes occur, how do we determine how the changes affect security?
  - Review, review, then review some more.

- DIACAP, -RMF for IS and PIT systems mandates continuous review process…
- Reviews require **time**, **expertise**, **manpower**, **money**.

RMF: 8510.01, March 2014



Step 2
SELECT
Security Controls

- Common Control Identification
- Develop system-level continuous monitoring strategy
- Review and approve the security plan and continuous monitoring strategy

Step 4
ASSESS
Security Controls

- Develop and approve Security Assessment Plan
- Assess security controls
- SCA prepares Security Assessment Report (SAR)
- Conduct initial remediation actions

Assess, review, remediate… rinse, repeat…

- Good in theory, but in practice? Everything is done manually; i.e. slowly.
- Cannot scale as complexity increases.
- Mobile? Cloud-based platforms?
- Constant change.
- Constantly increasing complexity.

What's the risk?

- Fast and loose: **data spills**.
  - Quick and dirty, miss critical faults.
- Slow and steady: **lose agility**.
  - Must avoid review "backlog mission impossible".
  - Adversaries will roll out new systems faster than us.
- Can't just throw more experts at the problem…
  - Brooks' Law.
  - Too many cooks! Increases accidental complexity.
  - "9 women can't make a baby in 1 month!"

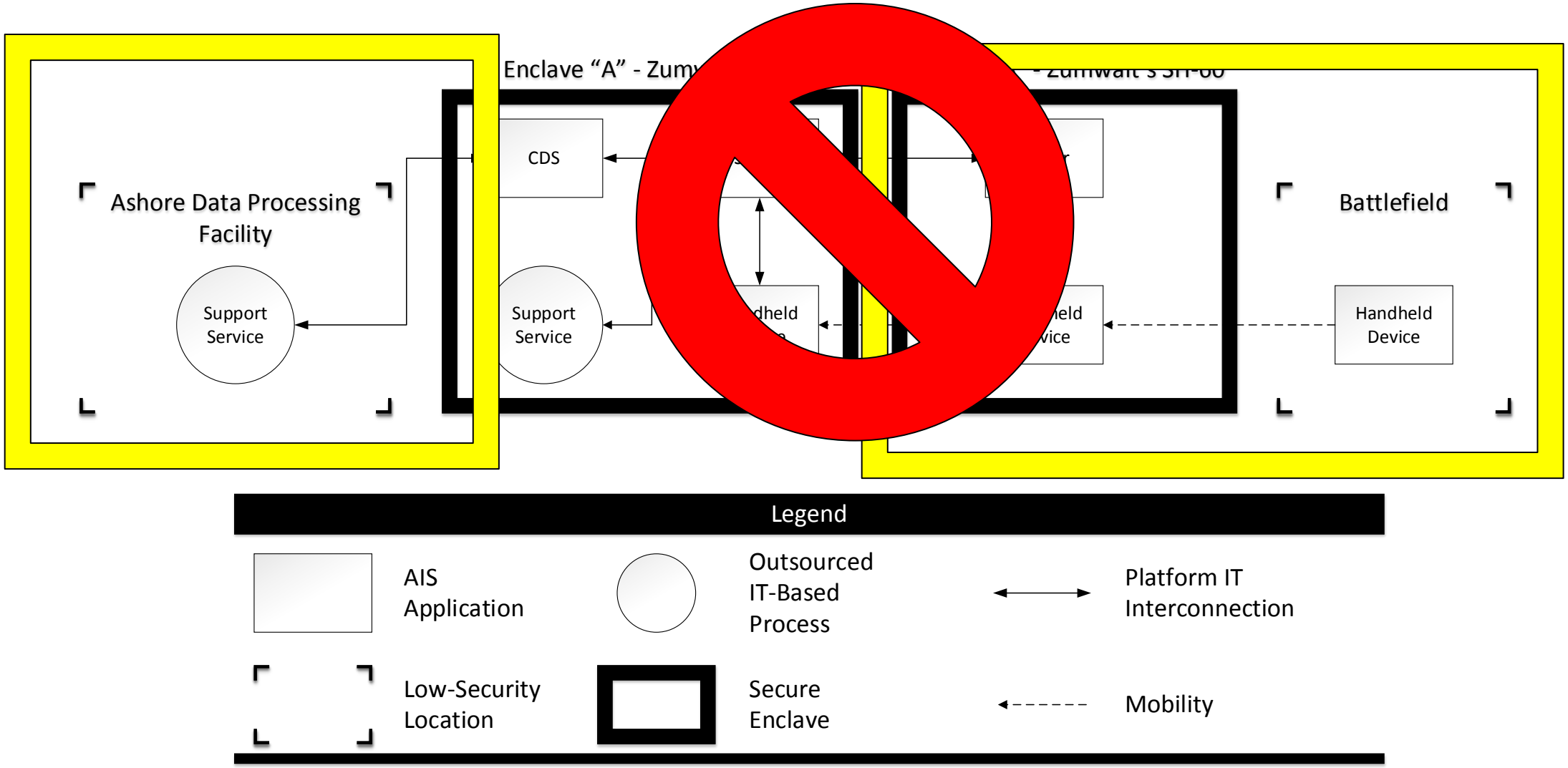
What kind of solution is needed?

- Use automation.
- Scale with evolving architectural assumptions.
- Do analysis computationally.
- Focus on adding new features, let the analysis determine the impact.

- **Result: Rapid analysis at recertification (or design) time.**

- Focus on the parts that commensurate with risk:
  - Data.
  - Secure enclave boundaries.
  - Changes.

What parts do we focus on?



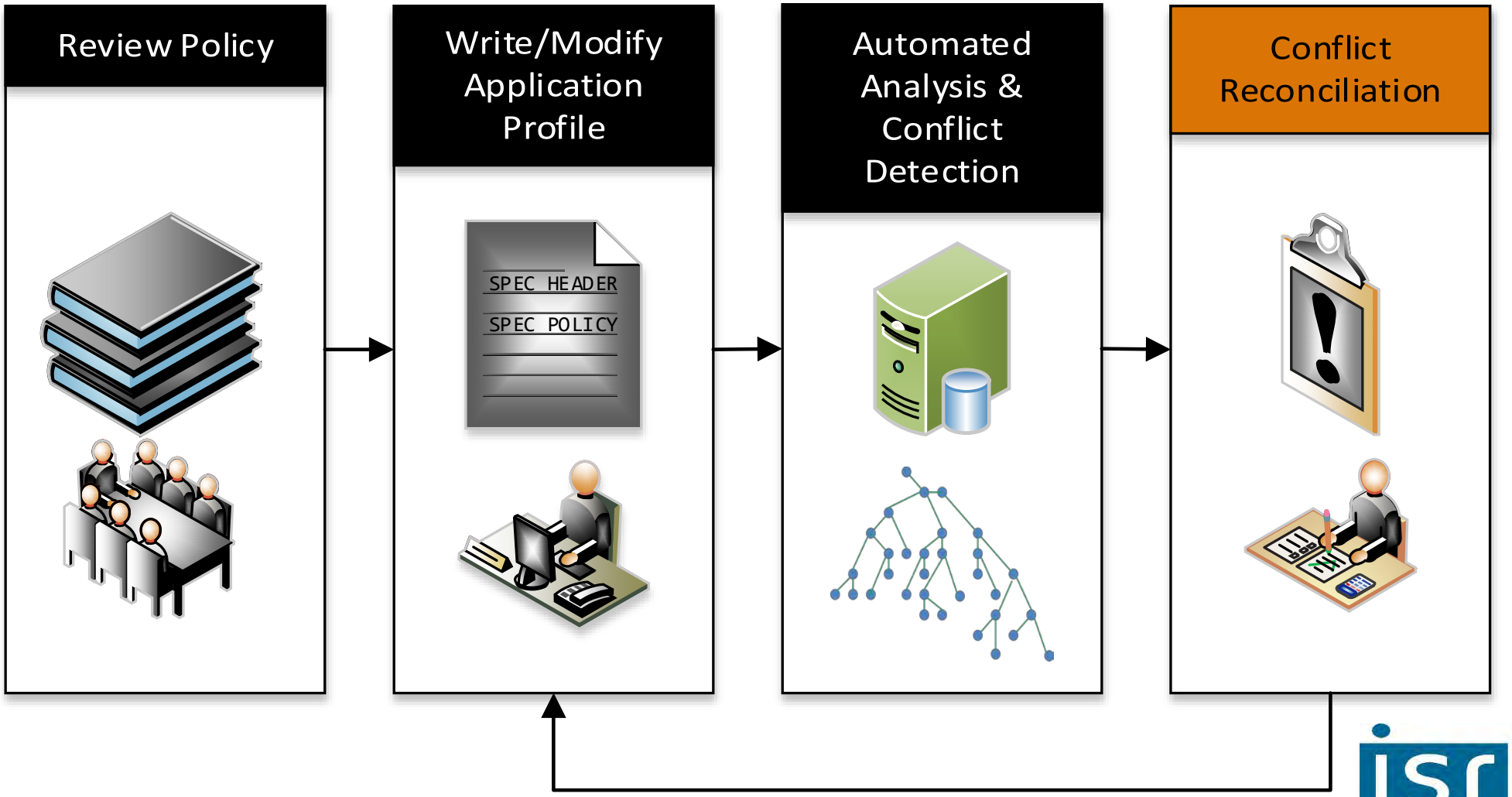
Technical Background

- Application Profile Language, model-checking.
- Semantic parameterization (Breaux et al., 2008)
  - Actions on data; actors, objects, purposes, source, destination.

- Bell-LaPadula: high-, low-confidentiality.
- Characterize the *purpose*; security level.
- Express compositions; logical subsumption.
  - Containment
  - Disjointness

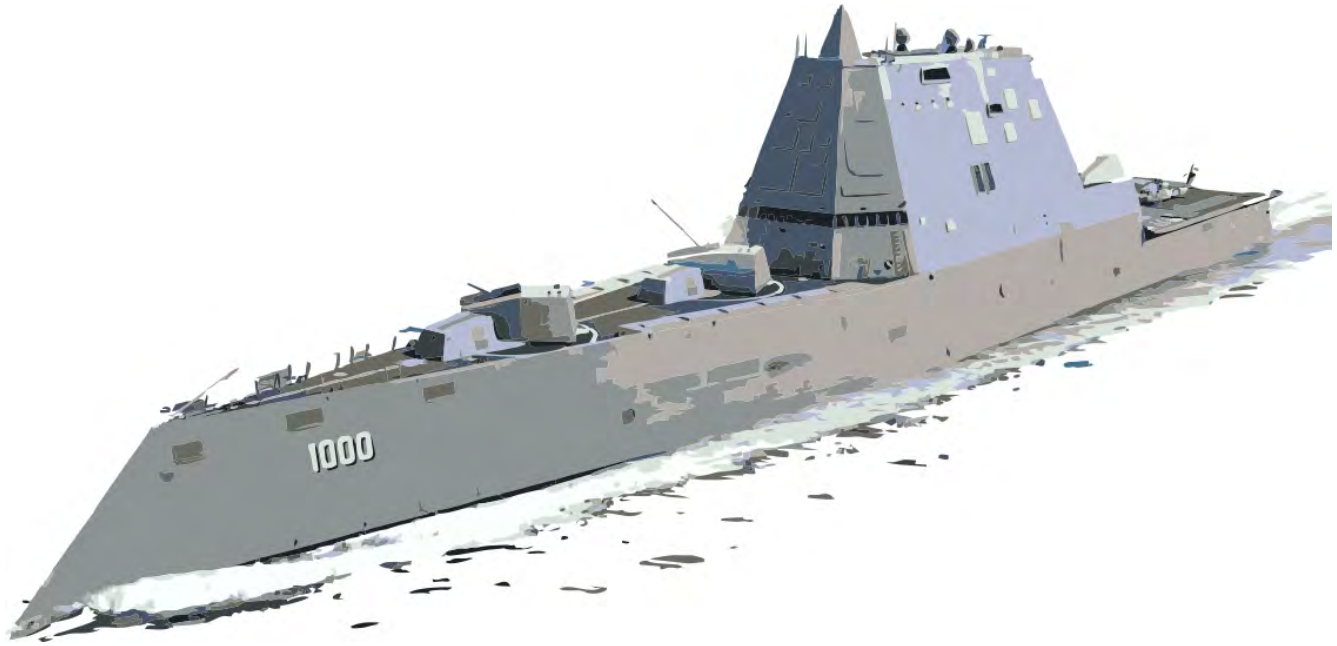
- **This forms the basis for our application profile language.**

Technical Background



Running Example

- Public accounts of real-world ship.
- Zumwalt-class destroyer.
- TSCE Infrastructure
- 6 MLOC
- Focus on software requirements:
  - Sensory and information sharing capabilities.



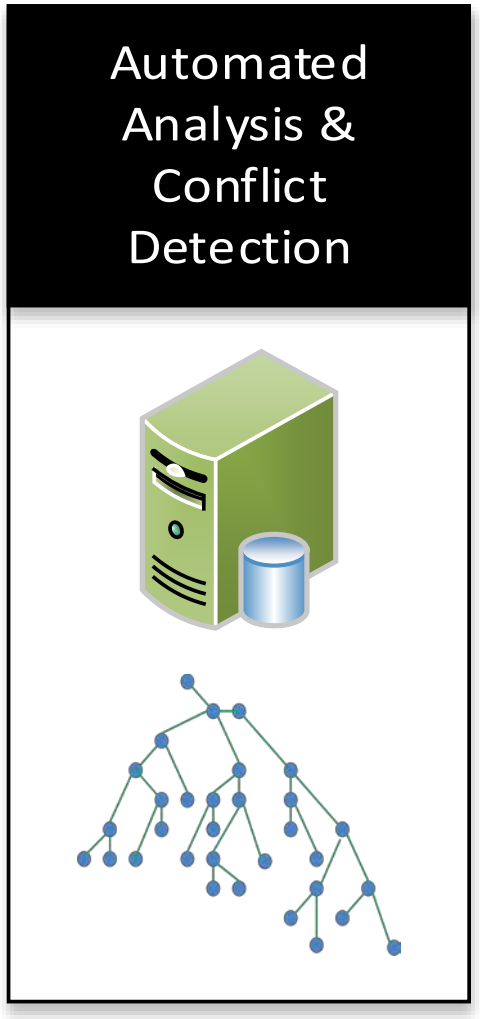

02-92-0-C
FR92-103
OP1
Surface Dominance - Undersea Warfare
Land Attack Warfare
Air Dominance

Approach

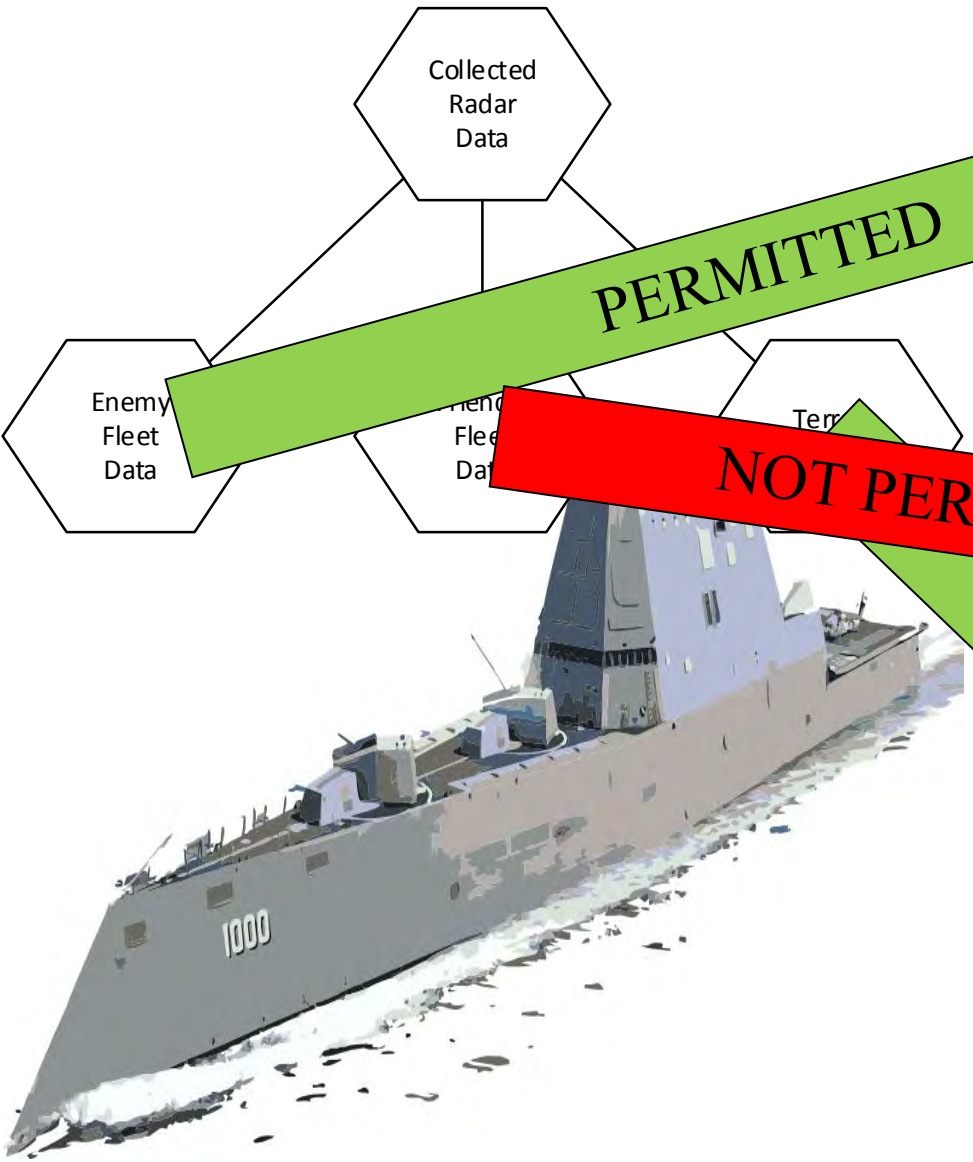
- Application profiles
  - Actions on data:
    - Collection
    - Use
    - Transfer
  - Traces:
    - Collection-Use
    - Collection-Transfer
    - Vice-versa

Approach

- Conflict Detection
  - Policy may specify a prohibition and a right on the same data, for the same purpose.
  - Leads to conflict.



D collected_radar_data < friendly_data, enemy_data, terrain_data



USS Zumwalt

PERMITTED

NOT PERMITTED

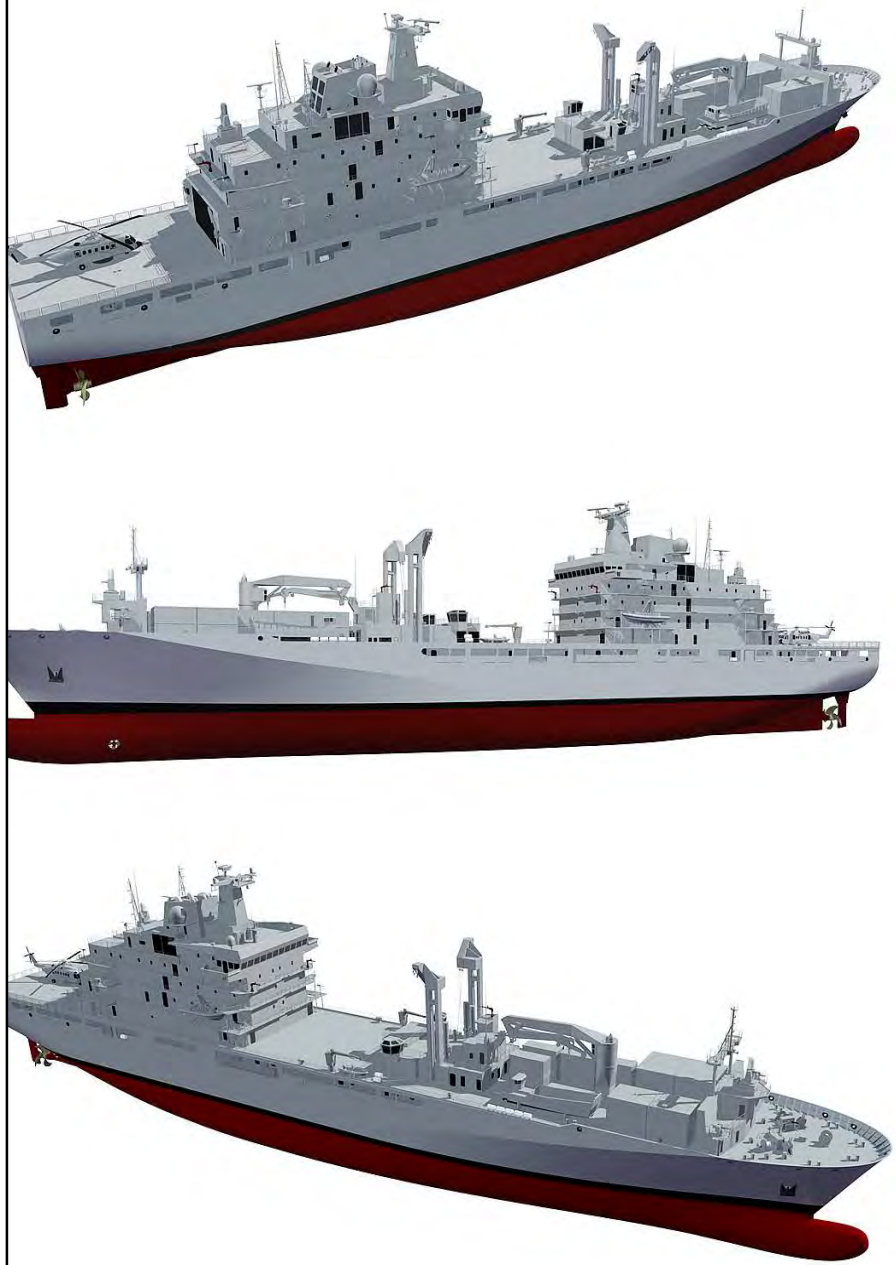
PERMITTED

Low Confidentiality

Enemy Fleet Data

Friendly Fleet Data

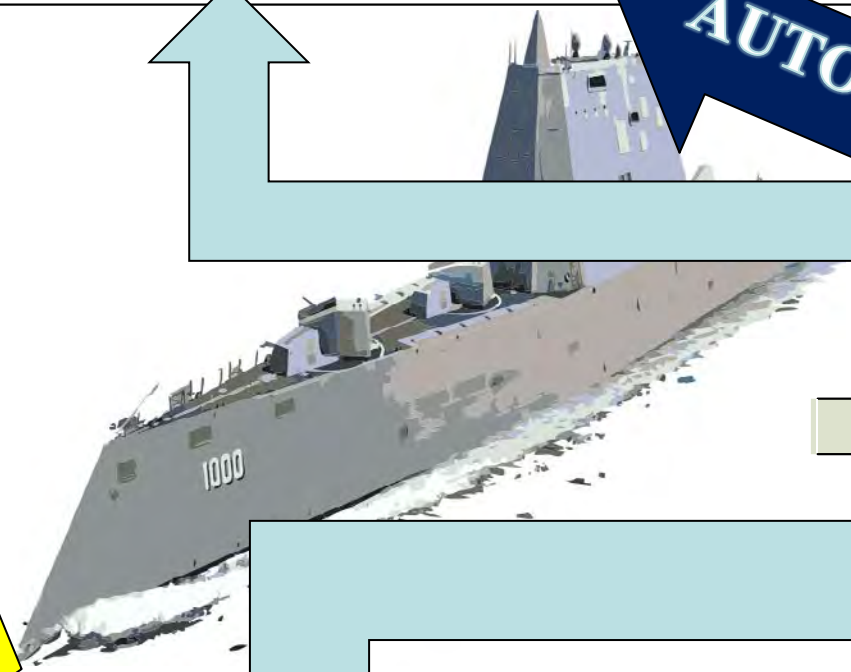
Terrain Data



Friendly Fleet

Data Definitions (Profile's Header)

```
SPEC HEADER
D collected_radar_data < friendl    ta, enemy_data, terrain_data
```



Radar System

S Zumwalt

Definitions (Profile's Policy)

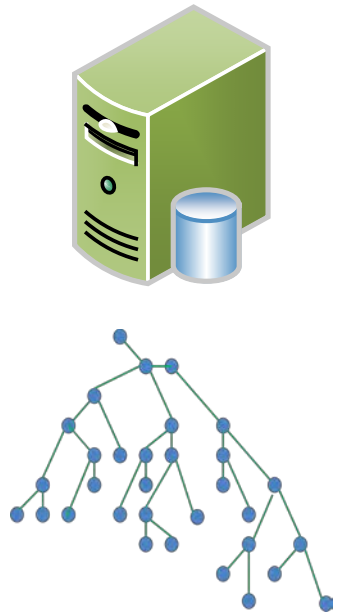
```
P COLLEC    ar_system FOR high_confidentiality
P TRANSFER    leet FOR low_confidentiality
P TRANSFER    a TO friendly_fleet FOR low_confidentiality
P TRANSFER    O friendly_fleet FOR high_confidentiality

R TRANSFER    ly_data TO anyone FOR low_confidentiality
```

AUTOMATION

AUTOMATION

AUTOMATION

Automated Analysis & Conflict Detection



1. Permit collection of collected radar data from Zumwalt's radar system, designating it as high-confidentiality data.

Application Profile Language	Formalization in Description Logic
`P COLLECT collected_radar_data FROM radar_system FOR high_confidentiality`	$T \models p_0 \equiv COLLECT \sqcap \exists \text{has}Object.$ collected_radar_data $\sqcap$ $\exists hasSource.$ radar_system $\sqcap$ $\exists hasPurpose.$ high_confidentiality

2. Permit transfer of data about enemy vessels to friendly fleet members for general, low-confidentiality purposes.

Application Profile Language	Formalization in Description Logic
`P TRANSFER enemy_data TO friendly_fleet FOR low_confidentiality`	$T \models p_1 \equiv$ TRANSFER $\sqcap \exists \text{has}Object.$ enemy_data $\sqcap$ $\exists has\text{Target}.$ radar_system $\sqcap$ $\exists hasPurpose.$ low_confidentiality

3. Permit transfer of all collected radar data to friendly fleet members for general, low confidentiality purposes. *This rule generates a conflict, which is explained below.*

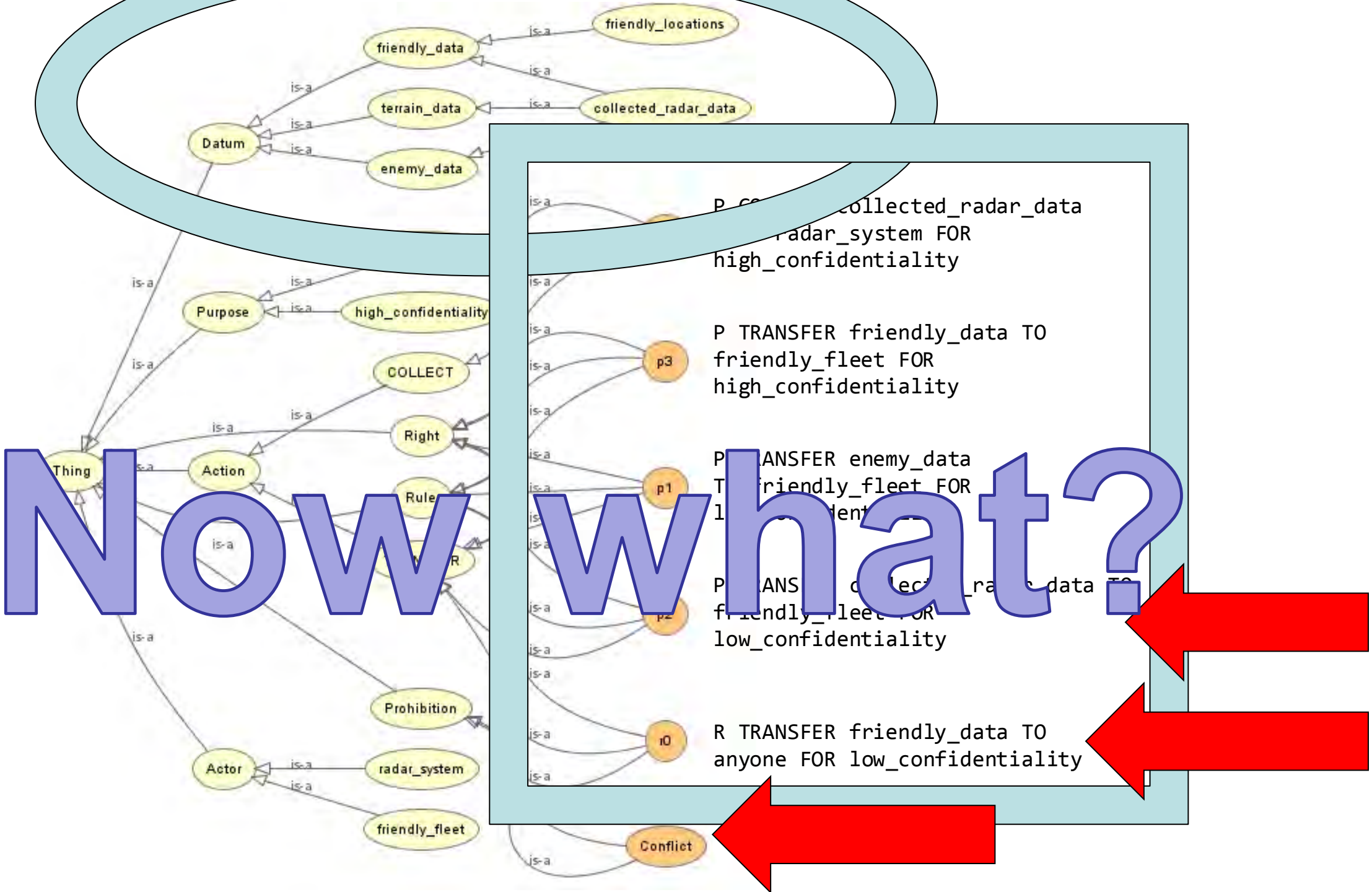
Application Profile Language	Formalization in Description Logic
`P TRANSFER collected_radar_data TO friendly_fleet FOR low_confidentiality`	$T \models p_2 \equiv$ TRANSFER $\sqcap \exists \text{has}Object.$ collected_radar_data $\sqcap$ $\exists has\text{Target}.$ friendly_fleet $\sqcap$ $\exists hasPurpose.$ low_confidentiality

4. Permit transfer of data about friendly vessels to friendly fleet members for specific, high-confidentiality purposes.

Application Profile Language	Formalization in Description Logic
`P TRANSFER friendly_data TO friendly_fleet FOR high_confidentiality`	$T \models p_3 \equiv$ TRANSFER $\sqcap \exists \text{has}Object.$ friendly_data $\sqcap$ $\exists has\text{Target}.$ friendly_fleet $\sqcap$ $\exists hasPurpose.$ high_confidentiality

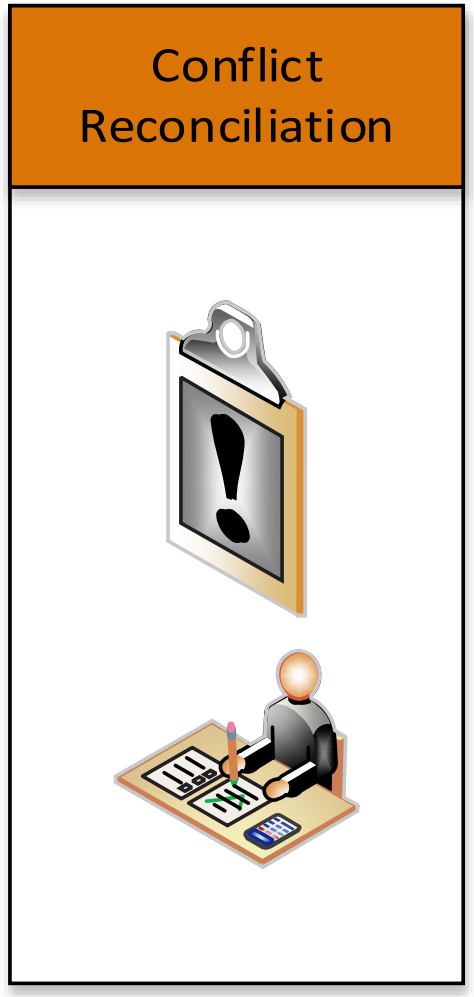
5. Prohibit transfer of friendly fleet data to anyone for general, low confidentiality purposes. *This rule conflicts with Rule 3, explained below.*

Application Profile Language	Formalization in Description Logic
`R TRANSFER friendly_data TO anyone FOR low_confidentiality`	$T \models r_0 \equiv$ TRANSFER $\sqcap \exists \text{has}Object.$ collected_radar_data $\sqcap$ $\exists has\text{Target}.$ Actor $\sqcap$ $\exists hasPurpose.$ low_confidentiality


friendly_locations
friendly_data
terrain_data
collected_radar_data
Datum
enemy_data
is-a
Purpose
high_confidentiality
COLLECT
Right
Thing
Action
Rule
Prohibition
Actor
radar_system
friendly_fleet
Conflict
P TRANSFER collected_radar_data
radar_system FOR
high_confidentiality
P TRANSFER friendly_data TO
friendly_fleet FOR
high_confidentiality
p3
p1
p2
p0
low_confidentiality
R TRANSFER friendly_data TO
anyone FOR low_confidentiality
Now what?

Reconciliation

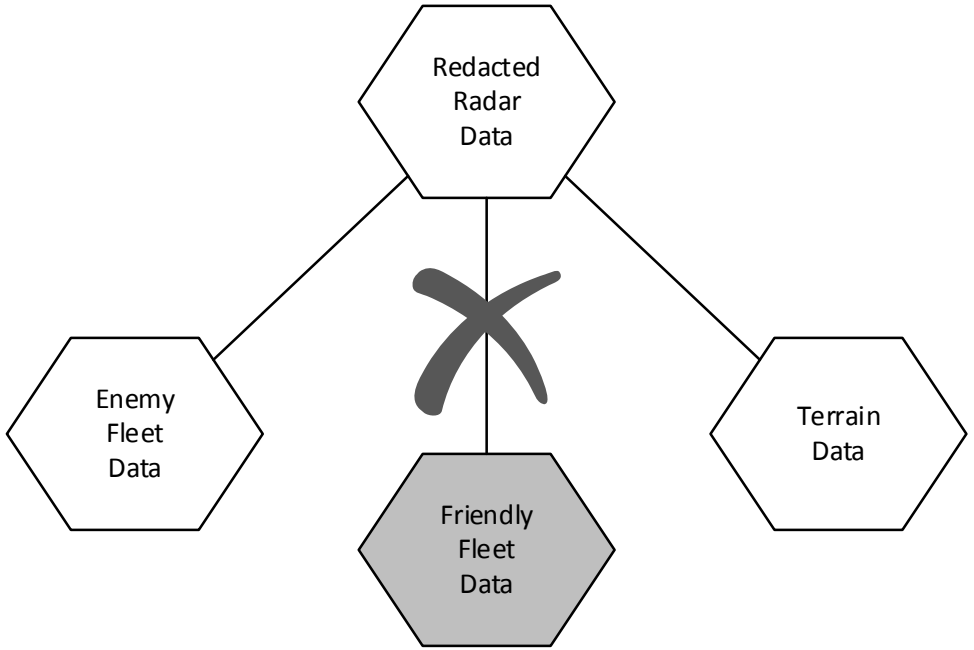
- Two reconciliation approaches identified:
  - Redaction
  - Generalization
- One approach that defeats these measures:
  - Merging



Redaction

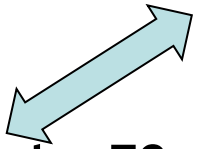
- Eliminate a subsumption relationship within a collection.
- Permits the new (redacted) collection to be used for low-confidentiality purposes.

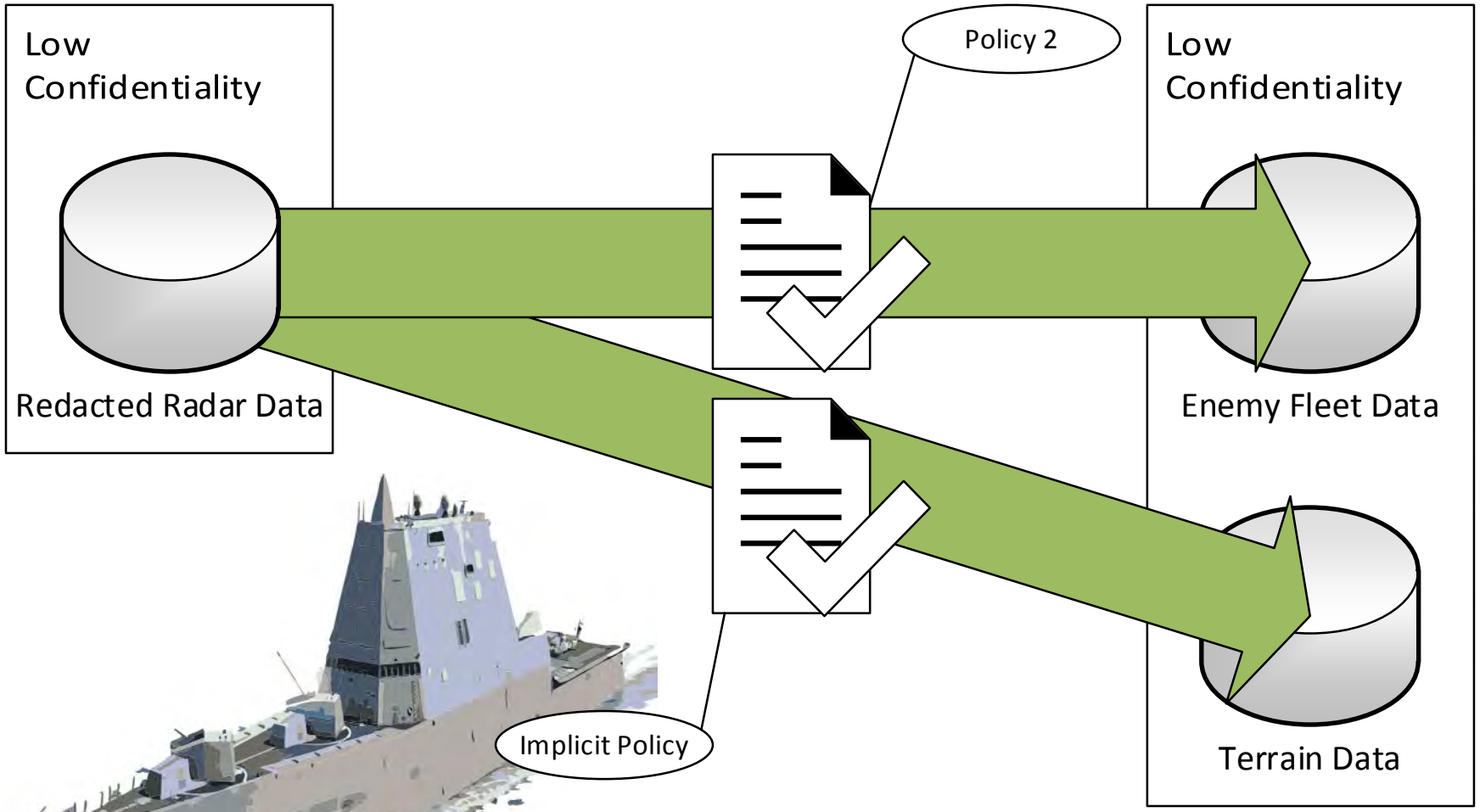
```
D redacted_radar_data <
enemy_fleet_data, terrain_data
```



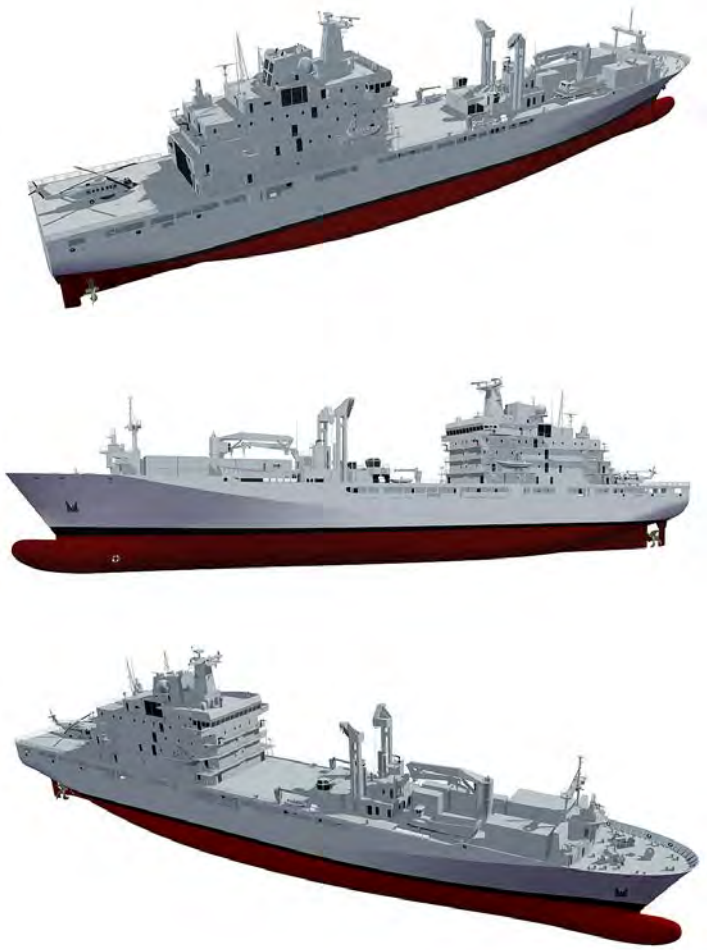
Redaction

```
     SPEC POLICY
1    P COLLECT collected_radar_data FROM radar_system FOR high_confidentiality
2    P TRANSFER enemy_data TO friendly_fleet FOR low_confidentiality

     REDACT(collected_radar_data -> redacted_radar_data, friendly_data,
     low_confidentiality)

3    P TRANSFER redacted_radar_data TO friendly_fleet FOR low_confidentiality
4    P TRANSFER friendly_data TO friendly_fleet FOR high_confidentiality

5    R TRANSFER friendly_data TO anyone FOR low_confidentiality
```





USS Zumwalt



Friendly Fleet

Generalization

- Some types of data can be **fuzzified**.
  - Add noise, decrease fidelity.
- Numerical data:
  - Coordinates, time…
- All collections' members must be generalized.

Merging

- Combine redacted data with un-redacted to recreate original.
- Combine generalized data with **de-noised** data to recreate original.

Distinguishing the Merging Risk

Policy Violation

1. Collect data for **high-confidentiality** purpose.
2. Collect other data for **low-confidentiality** purpose.
3. Repurpose high-confidentiality data, violate policy.

Merging

1. Collect data for **low-confidentiality** purpose.
   - Data is subset of redacted superset.
2. Collect related data for **low-confidentiality** purpose.
   - Data is negation of superset and redacted superset.
3. Merge two disjoint collections.

Similarly purposed data flows may be merged.

Merging Risk Mitigation

- Can catch merging risks as a result of conflict analysis.
  - Check subsumed purposes.
  - Trace data flows, transfer only what data is needed.

- Mitigates human error due to missed interpretations.

Recertification Triggers

How do you know when to run the analysis?

- Reconcile a conflict? Rerun, recheck.
- Add a new feature? Rerun, recheck.
- Modify the policy? Rerun, recheck.

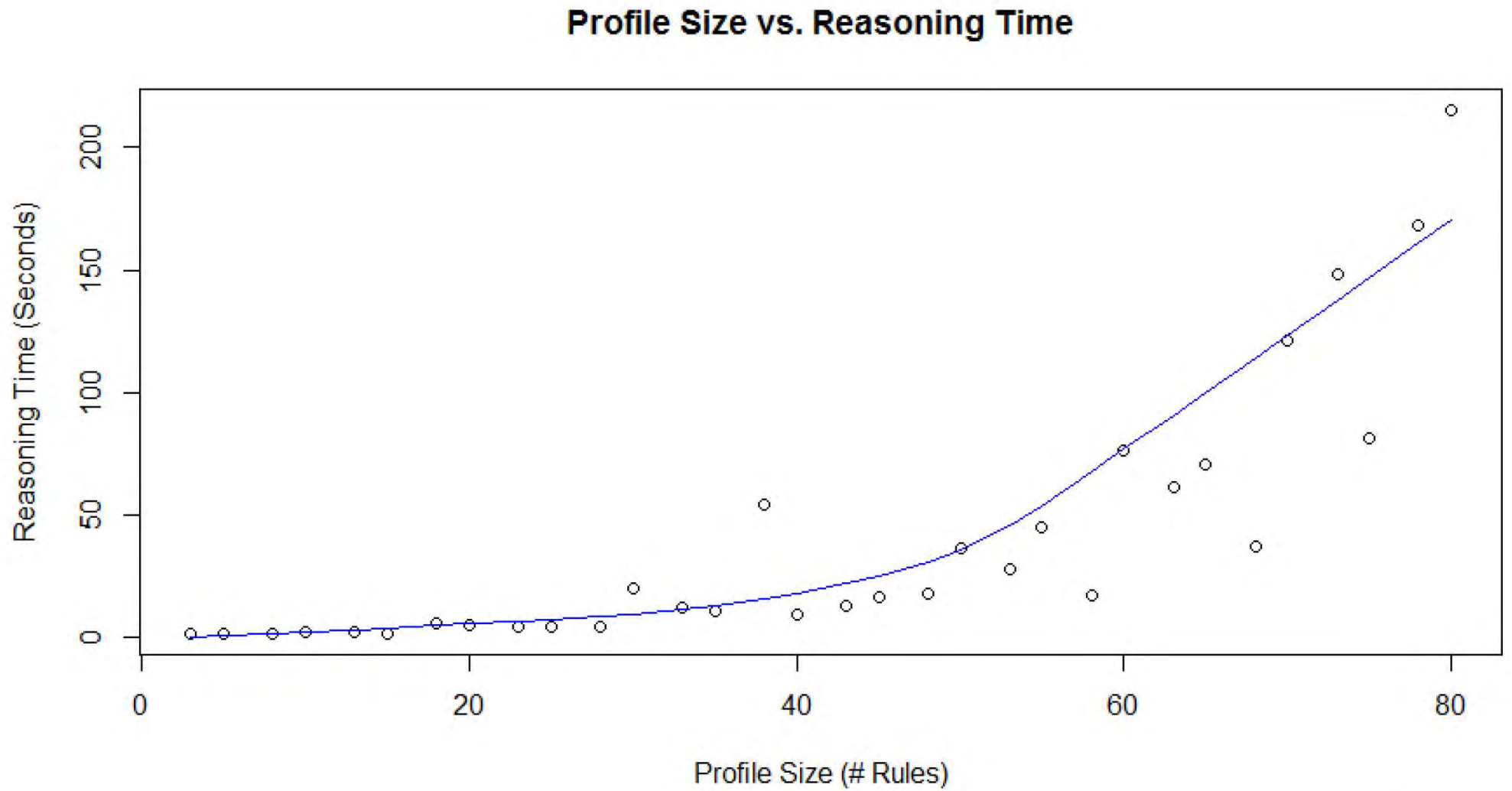
- Rapid analysis means recertification is rapid.

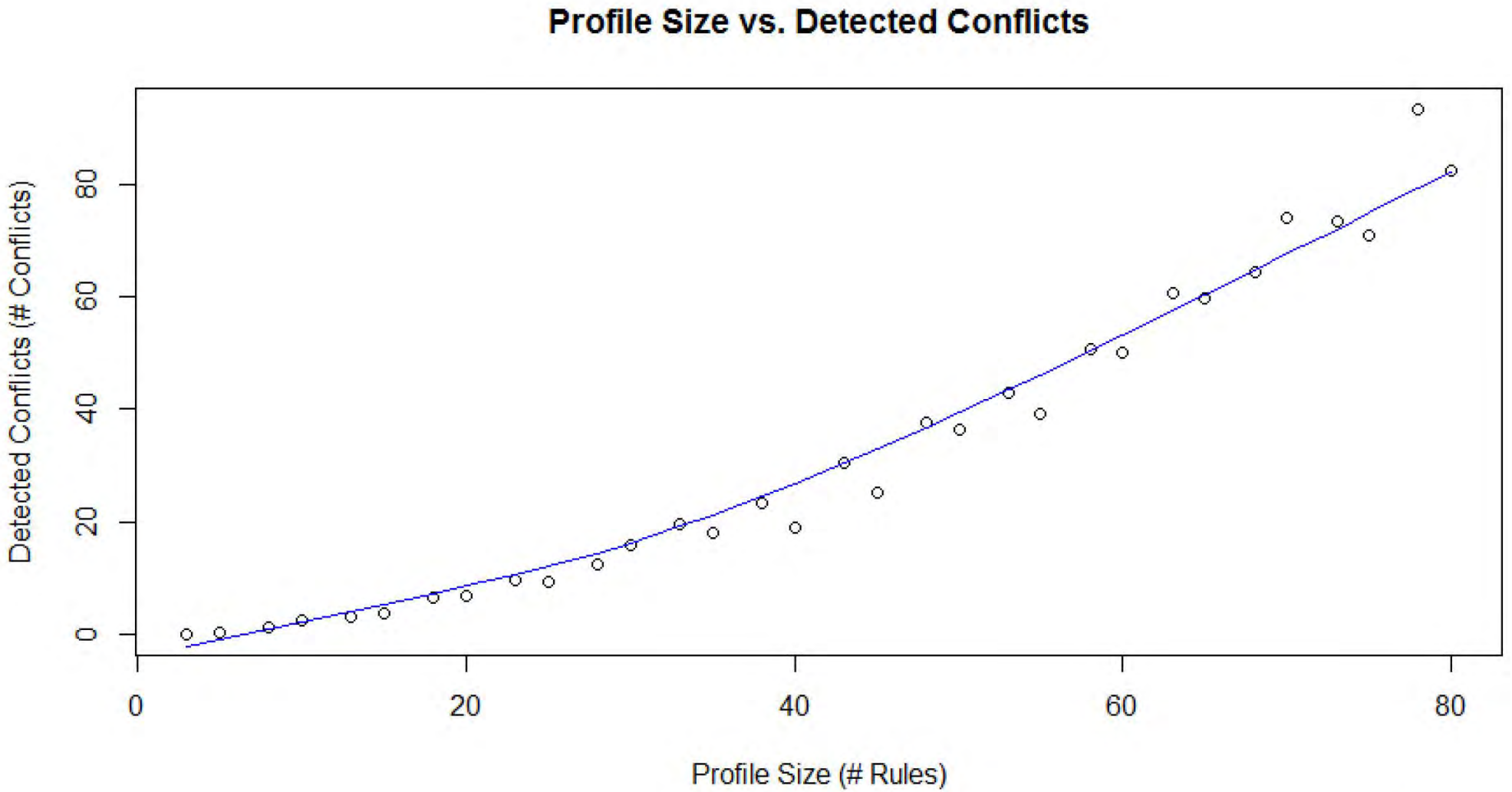
Does it scale?

- How fast can we do analysis? Is it fast enough to let us rerun whenever we want?

- Simulations; 27 repetitions, increasing number of rules [0-80], 1.13 conflicts per increasing rule.

No objective basis for comparison.


Profile Size vs. Reasoning Time
Reasoning Time (Seconds)
0
50
100
150
200
0
20
40
60
80
Profile Size (# Rules)


Profile Size vs. Detected Conflicts
Detected Conflicts (# Conflicts)
0
20
40
60
80
0
20
40
60
80
Profile Size (# Rules)

Does it scale?

- No statistically significant relationship between performance and number of conflicts.

$$\{\underline{r}(874) = .36, \underline{p} > .05\}$$

Average Profile Parsing Time	**<1 second**
Largest Profile Size	80 rules
Longest Profile Processing Time	**400 seconds**
Average Conflicts per Statement	1.13

Conclusions

- Yes, it scales:
  - Analysis can scale in quasilinear time.
- Simulations show that even huge profiles can be analyzed in roughly 7 minutes.

- What do we mean by huge profiles?
  - Hundreds of data flows.
  - Hundreds of rule combinations.
  - Hundreds of conflicts.

Future Work

- Extend automation to provide “hints” to analysts.
  - Profile development environment.
  - Automate reconciliation strategies.

- Characterize performance gain against manual processes.

Questions?

- Daniel Smullen

Graduate Research Assistant, Carnegie Mellon University

dsmullen@cs.cmu.edu

- Travis Breaux

Assistant Professor, Carnegie Mellon University

breaux@cs.cmu.edu